

## УДОСКОНАЛЕННЯ АЛГОРИТМУ ФОРМУВАННЯ ПАКЕТІВ ДАНИХ ДЛЯ УПРАВЛІННЯ БЕЗПЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ

*У статті запропоновано підхід до удосконалення алгоритму кодування, декодування пакетів даних на борту безпілотних літальних апаратів, що дозволить збільшити швидкодію цих процесів та проводити обробку і формування інформаційних пакетів у реальному масштабі часу.*

**Постановка проблеми.** У сучасних автоматичних системах управління безпілотними літальними апаратами (БПЛА) алгоритми кодування та декодування даних не є оптимальними за обчислювальними і системними затратами та вимагають повного приймання усієї інформаційної послідовності й запису її в масив статичного оперативного запам'ятовуючого пристрою перед початком обробки пакета даних [1]. Враховуючи надлишковість інформаційних пакетів, що циркулюють у системі управління БПЛА, такий підхід вимагає значних системних обчислень та затрат ресурсів оперативного запам'ятовуючого пристрою, що може бути критичним та призведе до збоїв системи управління БПЛА [1, 2]. У зв'язку з цим актуальною є задача, розв'язання якої передбачає прискорення процесів кодування, декодування даних шляхом зменшення інформаційного надлишку пакетів та пришвидшення обчислень.

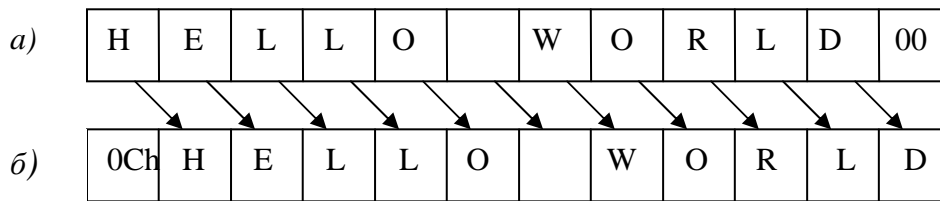
**Огляд останніх досліджень і публікацій.** На сьогоднішній день найбільш поширеними алгоритмами кодування та декодування пакетів даних є SLIP, PPP, AX.25, COBS, робота яких полягає в заміні “зарезервованих” чи “заборонених” значень за умов неперевищення середнього значення довжини пакета [1–4]. Вказані алгоритми не оптимальні в обчислювальному відношенні та можуть генерувати інформаційний надлишок при формуванні пакета до 100% від його початкового розміру, що збільшує час обробки даних на борту БПЛА. Крім того, процеси кодування, декодування даних за даними алгоритмами розпочинаються тільки після надходження усього пакета даних, що є критичним для системи управління БПЛА [5, 6]. Найменше обчислювальних потужностей потребує алгоритм COBS, але відсутність можливості обробки даних у реальному масштабі часу стримує його застосування на борту БПЛА [7–11].

**Формулювання завдання дослідження.** З огляду на вимогу швидкого декодування та кодування пакетів даних під час управління динамічними об'єктами, такими як БПЛА, необхідність використання при цьому пристроїв, побудованих на мікроконтролерах та програмованих логічних схемах, що мають обмежені системні ресурси й обчислювальні можливості, а також низьку, порівняно з персональними обчислювальними машинами, швидкодію, **метою статті** є вдосконалення алгоритму COBS у напрямку скорочення часу

кодування та декодування даних за рахунок зменшення інформаційного надлишку пакетів даних, їх формування та обробки в реальному масштабі часу.

**Виклад основного матеріалу.** В основі роботи алгоритму COBS лежить ідея виключення із потоку даних усіх байтів, що мають певне значення, яке використовують у подальшому як маркер без втрати однозначності. Значення може бути будь-яким, але частіше використовують нульове [10].

Приклад кодування символічних даних алгоритмом COBS з виключенням нульових послідовностей наведено на рис. 1.

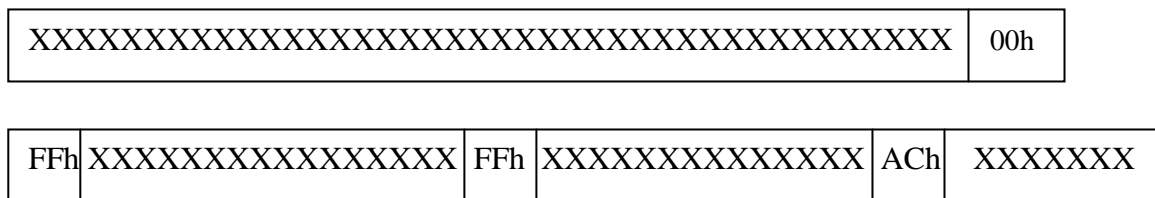


*Рис. 1. Кодування даних алгоритмом COBS:*

*a) незакодований пакет даних; б) закодовані дані на передачу*

Величина 0Ch вказує на порядковий номер нуля в послідовності до кодування та одночасно є маркером початку пакета. Таким чином, кодування даних є більш ефективним за рахунок зменшення надлишковості, але для визначення маркера початку пакета необхідно оцінити весь пакет чи деяку кількість даних, доки не зустрінеться нуль у послідовності, що вимагає додаткового часу та резервування буфера в оперативному запам'ятовуючому пристрої. Як видно з прикладу, при передачі символічних даних усі їх кодові значення відмінні від нуля, тому ефективність алгоритму для такої передачі буде низькою.

Розглянемо роботу алгоритму за умови розміру пакета 680 байт, де лише один нульовий, а кодування здійснюється трьома блоками COBS-алгоритму (рис. 2).



*Рис. 2. Закодований пакет даних*

Перші два блоки мають максимальну довжину та не містять нуля. Останній є стандартним блоком з 171 байтом даних і останнім нулем. Це означає, що 680 байт даних закодовано в 682 байти, а надлишок становить до 0,3%.

Приклад кодування та декодування даних у шістнадцятковій системі обчислення наведено на рис. 3.

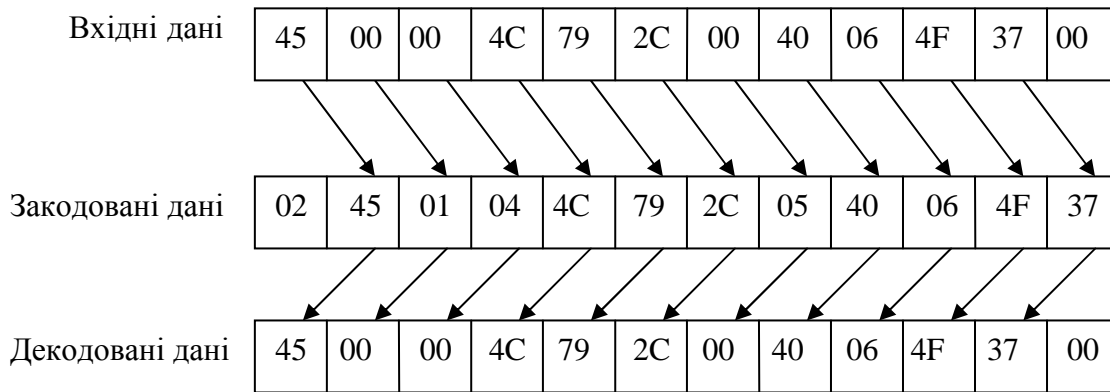


Рис. 3. Вхідний та вихідний пакети даних COBS-алгоритму

Як видно з рис. 3, для кодування даних необхідно попередньо обробити кількість байтів, рівну кількості ненульових байтів, що призводить до збільшення часу кодування та може бути неприйнятним для бортового обладнання БПЛА. З метою уникнення цього недоліку запропоновано алгоритм кодування реалізувати за схемою, наведеною на рис. 4.

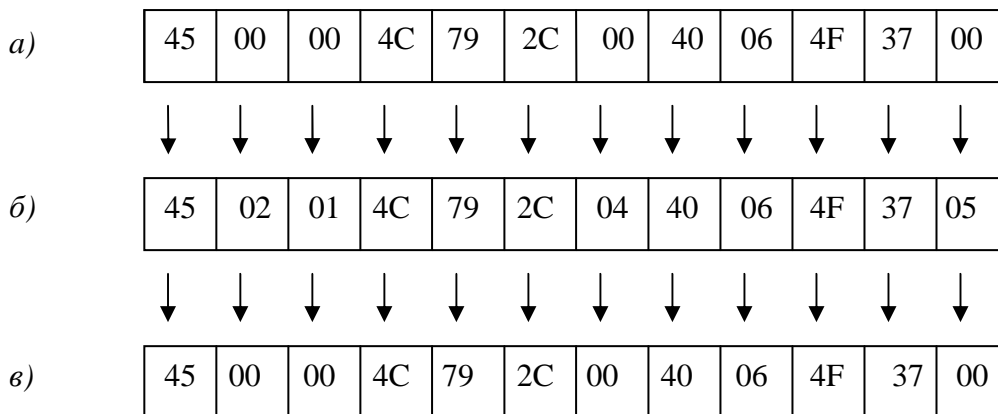
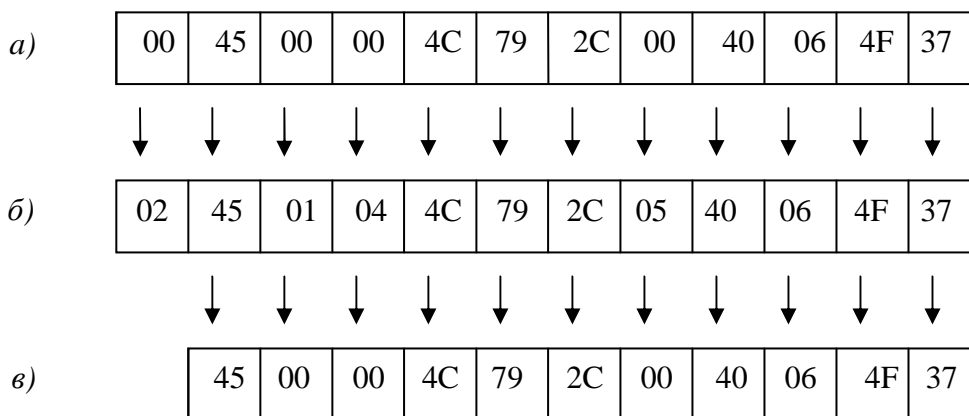


Рис. 4. Схема алгоритму кодування пакета на борту БПЛА:

а) незакодований пакет даних; б) закодовані дані на передачу; в) розкодований пакет

Особливістю такого підходу є формування пакета (рис. 4б) передачі байтів даних та заміною першого нульового байта на його порядковий номер у пакеті. Наступні нульові байти замінюють значеннями кількості байтів, які були передані після попереднього нуля. У випадку розташування декількох нульових байтів один за одним їх замінюють значенням 01h. Останній байт даних завжди міститиме значення порядкового номера останнього нульового байта, і тому такий пакет розкодовують з кінця. Формування пакета за запропонованою схемою не потребує значних обчислювальних та системних затрат, а отже, має значну швидкодію. Декодування цього пакета проводять з кінця, це передбачає прийом всього пакета даних, що для наземних станцій управління БПЛА є некритичним.

Для передачі даних на борт БПЛА, апаратура якого є критичною щодо швидкості декодування пакетів, запропоновано використовувати алгоритм, який реалізується за схемою, наведеною на рис. 5. Декодування такого пакета починається з першого байта, не потребує прийому всього пакета, здійснюється в реальному масштабі часу й закінчується із прийняттям усього пакета даних.



*Рис. 5. Схема алгоритму кодування пакета для передачі на борт БПЛА: а) незакодований пакет даних; б) закодовані дані; в) розкодований пакет*

Таким чином, кодування даних за схемами, наведеними на рис. 4, 5, дозволяє обробляти дані на борту БПЛА в реальному масштабі часу.

Для оцінювання надлишковості інформаційного пакета, закодованого за удосконаленим алгоритмом, візьмемо до уваги той факт, що кількість байтів вхідної послідовності може бути різною, тому для обчислення надлишку інформації середню кількість вихідних байтів поділимо на середню кількість вхідних байтів. Якщо перший байт незакодованого повідомлення є нульовим, то цей байт кодують як цілий блок, а на виході алгоритму отримуємо один байт. Імовірність  $p(01h)$  такого випадку є  $256^{-1}$ . Коли в послідовності, що підлягає кодуванню, нульовим є другий байт, на виході алгоритму отримаємо два байти, а ймовірність  $p(02h)$  такого випадку становить  $\frac{255}{256^2}$ . Імовірність

$p(n)$ , що у вихідній послідовності кількість ненульових байтів, які слідує за нульовим, дорівнює  $n-1$  ( $n \leq 254$ ) і становить  $\left(\frac{255}{256}\right)^{n-1} \times \left(\frac{1}{256}\right)$ .

Найдовшим пакет може бути тоді, коли після 254 байтів не було жодного нульового. Імовірність  $p(FFh)$  такого випадку  $\left(\frac{255}{256}\right)^{254}$ . Коефіцієнт відношення середнього значення вихідних байтів до середнього значення вхідних байтів розраховуємо за таким виразом:

$$\frac{\left(\sum_{n=1}^{254} np(n)\right) + 255p(255)}{\left(\sum_{n=1}^{254} np(n)\right) + 254p(255)} \approx 1,002295.$$

Таким чином, теоретичний надлишок інформації для випадкової послідовності даних не буде перевищувати 0,23%, що в середньому втричі менше від значення інформаційних надлишків для алгоритмів SLIP, PPP і AX.25.

Для перевірки вдосконаленого алгоритму було виготовлено пристрій кодування й декодування пакета даних на основі мікроконтролера АТmega64 та проведено експеримент, у результаті якого було встановлено його працездатність при кодуванні даних великого об'єму та відповідність показників ефективності теоретично розрахованим.

**Висновки.** Удосконалений алгоритм зберігає характеристики COBS-алгоритму та має перевагу за швидкістю, що дозволяє формувати та обробляти пакети даних. Упровадження вдосконаленого алгоритму в процес кодування, декодування даних, що циркулюють між наземною станцією та БПЛА, дозволить: підвищити ефективність і надійність управління останніми, знизити вимоги до апаратної частини систем управління та зменшити кількість нештатних ситуацій.

Напрямок подальших досліджень є створення нових протоколів на основі вдосконаленого алгоритму COBS з метою використання в мережах пакетного радіо при розробці протоколів для формування кадрів передачі даних.

### СПИСОК ЛІТЕРАТУРИ

1. Stuart Cheshire. Consistent Overhead Byte Stuffing / Stuart Cheshire, Mary Baker // IEEE / ACM Transactions on networking. – April 1999. – Vol.7, NO. 2. – P. 110–122.
2. Pierre Humblet. Generalization of Huffman Coding to Minimize the Probability of Buffer Overflow / Pierre Humblet // IEEE Transactions on Information Theory. – March 1981. – Vol. IT-27. – P. 230–232.
3. A. Artom. Choice of Prefix in Self-Synchronizing Codes / A. Artom // IEEE Transactions on Communications. – April 1972. – Vol. COM-20. – P. 253–254.
4. R. Barker. Group Synchronizing of Binary Digital Systems Communication / R. Barker, W. Jackson. – New York : Academic Press, September 1953. – 302 p.
5. Chris Bennett. The Overheads of Transnetwork Fragmentation / Chris Bennett // Computer Networks. – February 1982. – Vol. 6. – P. 21–36.
6. H. W. Braun. Web Traffic Characterization: an assessment of the impact of caching documents from NCSA's web server / H. W. Braun and K. Claffy. – Chicago, October 1994. – 17 p.
7. D. E. Knuth. Dynamic Huffman Coding / D. E. Knuth // Journal of Algorithms. – 1985. – Vol. 6. – P. 163–180.
8. James Carlson. Principal Software Engineer / James Carlson // IronBridge Networks. – September 1997. – P. 121–136.
9. Stuart Cheshire. Experiences with a Wireless Network in MosquitoNet / Stuart Cheshire and Mary Baker // IEEE Micro. – February 1996. – P. 53–59.
10. Stuart Cheshire. Consistent Overhead Byte Stuffing / Stuart Cheshire and Mary Baker // Proceedings of ACM SIGCOMM 1997. – September 1997. – P. 123–127.
11. Stuart Cheshire. Consistent Overhead Byte Stuffing : Ph.D. dissertation / Stuart Cheshire. – California, Stanford : Stanford University, March 1998. – 197 p.

Подано 14.05.2015

**В. А. Катюха, А. М. Токарь**

**УСОВЕРШЕНСТВОВАНИЕ АЛГОРИТМА ФОРМИРОВАНИЯ ПАКЕТОВ ДАННЫХ ДЛЯ УПРАВЛЕНИЯ БЕСПИЛОТНЫМИ ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ**

*В статье предложен подход к усовершенствованию алгоритма кодирования, декодирования пакетов данных на борту беспилотных летательных аппаратов, что позволит увеличить быстродействие этих процессов и проводить обработку и формирование информационных пакетов в реальном масштабе времени.*

**V. A. Katyukha, A. N. Tokar**

**IMPROVEMENT OF THE ALGORITHM OF THE SHAPING PACKAGE GIVEN FOR GOVERNING UNMANNED FLYING VEHICLE**

*In article is offered to improvement of the algorithm of the coding, decoding package on boards unmanned flying vehicles that will allow to enlarge speed of these processes and conduct processing and shaping information package in real scale of time.*